في الكثير من الاحيان لا يشترط على المهندس استخدام نوع محدد من المايكروكنترولر ... فهناك الكثير من الانواع التي تتفوات في قدراتها وامكانياتها مثل وجود ذاكره أكبر (ROM أو RAM) أو عدد أطراف اكبر أو خاصيه أو مجموعة من الخواص موجوده في نوع دون وجودها في نوع آخر مثل USB , PWM, EEPROM , UART وغيرها ...

ولكن للأسف الشديد عندما يكون المبرمج معتاد على التعامل مع نوع معين من المايكروكنترولر ، وينتقل الى نوع اخر قد يكون لسبب من الاسباب التي ذكرناها سابقاً ... يتفاجأ أن هناك أمور لا تعمل معه بشكل جيد .... حينها يعتقد ان هناك خلل في الكود الذي أنشأه أو ان المايكروكنترولر الذي قام بشراءه هو تالف .

لكن يكون السبب في ذلك هو جهل الشخص في التعامل مع هذا المايكروكنترولر الجديد لان غالباً ما تكون اطراف المايكروكنترولر مشتركه لتقوم بعدد من الوظائف ... لذلك في بداية هذه المحاضره سيكون حديثنا عن التعامل مع أي نوع من المايكروكنترولر

وفي الواقع ومن خلال الخبره ان اسهل طريقه تجعلك تبدأ مع أي مايكروكنترولر PIC جديد هي الداتاشيت الخاصه بهذه الـ PIC وتحديداً القسم الذي يحتوي على I\O Ports وقراءة المثال المكتوب بعنوان INITALIZING PORTA ويأتي دور الخبره الفعليه في ترجمة هذا الكود والذي يكون مكتوب بلغة الاسمبلي الى اللغه التي تحب التعامل معها والتي كانت في دورتنا هي لغة المايكروسي

أرى انك أصبحت قلقاً بشأن لغة الاسمبلي ..... لاتقلك بتأكيد ان المبرمج المتمكن لابد له من الاطلاع على لغة الاسمبلي لانها هي اساس اللغه التي يفهمها حتى المايكروبروسيسر والموجود داخل المايكروكنترولر.... لكن هدفنا في هذه الدوره هو توصيل المعلومه لك بشكل مبسط وميسر ... لهذا سأعلمك ما تحتاج معرفته من لغة الاسبلي لغايات القدره على التعامل مع الـ PORT للمايكروكنترولر الذي لم تتعامل معه مسبقاً .

وأكثر الـ PORT التي قد تسبب لنا الازعاج هو PORTA فمثلاً لو قمنا بفتح الداتا شيت على المثال الذي تحدثنا عنه سابقاً كما في الصورة التالي

The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 3-1: INITIALIZING PORTA

```
BCF     STATUS, RP0   ;
BCF     STATUS, RP1   ; Bank0
CLRF    PORTA         ; Initialize PORTA by
                      ; clearing output
                      ; data latches
BSF     STATUS, RP0   ; Select Bank 1
MOVLW   0x06          ; Configure all pins
MOVWF   ADCON1        ; as digital inputs
MOVLW   0xCF          ; Value used to
                      ; initialize data
                      ; direction
MOVWF   TRISA         ; Set RA<3:0> as inputs
                      ; RA<5:4> as outputs
                      ; TRISA<7:6>are always
                      ; read as '0'.
```

ما يهمنا هنا هو الامر MOVLW فعندما نرى هذا الامر وبجواره رقم ما ، فكأنما نقول للمايكروكنترولر خزن هذا الرقم في عقلك

وعندما نرى الامر MOVWF وبجواره اسم مسجل Register فكأنما نقول للمايكروكنترولر خزن القيمه الموجوده بعقلك في المسجل الفلاني

وأيضاً عندما نرى الامر CLRF وبجواره اسم مسجل ، فهذا يجعل قيمة المسجل تساوي صفر .... والخبره في البرمجه الان تكمن في تحويل هذه الاسس البرمجيه الازم التعامل معها في PORTA من لغة الاسمبلي الى لغة المايكروسي

وعملية التحويل كما يلي:

The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 3-1: INITIALIZING PORTA

```
BCF     STATUS, RP0   ;
BCF     STATUS, RP1   ; Bank0
CLRF    PORTA         ; Initialize PORTA by
                      ; clearing output
                      ; data latches
BSF     STATUS, RP0   ; Select Bank 1
MOVLW   0x06          ; Configure all pins
MOVWF   ADCON1        ; as digital inputs
MOVLW   0xCF          ; Value used to
                      ; initialize data
                      ; direction
MOVWF   TRISA         ; Set RA<3:0> as inputs
                      ; RA<5:4> as outputs
                      ; TRISA<7:6>are always
                      ; read as '0'.
```

ADCON1=0x06;

TRISA=0xCF;

والان ان ما شرحناه سابقاً كان عن PIC16f877a ولنتأكد من أن هناك اختلاف في التعامل بين نوع من المايكروكنترولر ونوع اخر لننظر الى المثال التالي وهو مأخوذ من الداتاشيت الخاصه بـ PIC16f628

EXAMPLE 5-1: INITIALIZING PORTA

```
CLRF    PORTA       ;Initialize PORTA by
                    ;setting
                    ;output data latches
MOVLW   0x07        ;Turn comparators off and
MOVWF   CMCON       ;enable pins for I/O
                    ;functions
BCF     STATUS, RP1
BSF     STATUS, RP0 ;Select Bank1
MOVLW   0x1F        ;Value used to initialize
                    ;data direction
MOVWF   TRISA       ;Set RA<4:0> as inputs
                    ;TRISA<5> always
                    ;read as '1'.
                    ;TRISA<7:6>
                    ;depend on oscillator
                    ;mode
```

لاحظ ان لأستخدام الـ PORTA من الـ PIC16f628 لابد من كتابة الجملة التالية

```
CMCON=0x07;
```

وهذا يكافأ الجملتين المحاطات باللون الاحمر في الصورة المجاورة

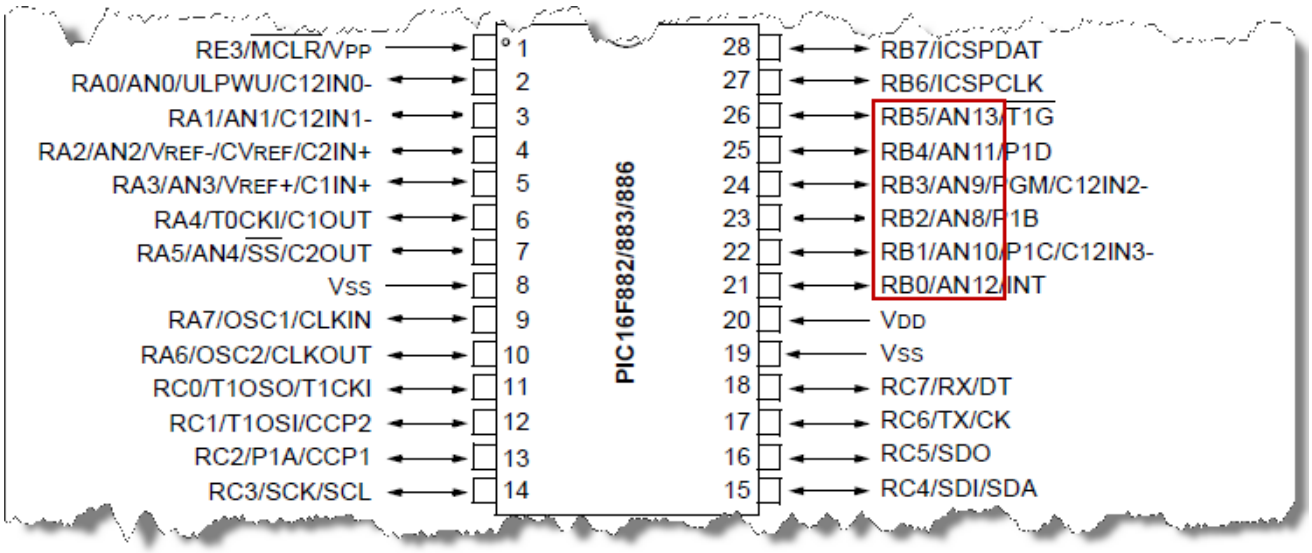
اما ان كنت تريد ان تستخدم PIC16F882/883/884/886/887 اذا انظر الى الصورة التاليه وهي مأخوذه ايضاً من الداتا شيت الخاصه بهذه الانواع من المايكروكنترولر

EXAMPLE 3-1: INITIALIZING PORTA

```
BANKSEL PORTA      ;
CLRF    PORTA      ;Init PORTA
BANKSEL ANSEL      ;
CLRF    ANSEL      ;digital I/O
BANKSEL TRISA      ;
MOVLW   0Ch        ;Set RA<3:2> as inputs
MOVWF   TRISA      ;and set RA<5:4,1:0>
                   ;as outputs
```

لـــن اعطيك الكود بلغة المايكـــــروسي سأترك لك المجال لتحاول بذلك لوحد مع ان هذا يعتبر تمـــــريناً بسيط نـــوعاً ما وخاصه بتـوضيح ذلك فـي الصـــــورة المجاوره

دعنا نتعمق قليلاً في الانواع الاخيره من المايكروكنترولر وهي PIC16F882/883/884/886/887 قم بفتح الداتاشيت الخاصه بهذه الانواع وانظر الى الـ Pin Diagram لها ..... وها هي في الصورة التاليه ..... انظر جيداً الا ترى ان هناك شيء غريباً ولم نعتد عليه ؟؟



لقد اعتدنا في الامثله السابقه ان مداخل الـ Analog هي عادتاً موجوده على PORTA ولكن هنا انظر الى PORTB ماذا تلاحظ ؟؟

اذا لابد أيضاً من النظر الى الـ PIN Diagram للمايكروكنترولر الذي سنتعامل معه ....

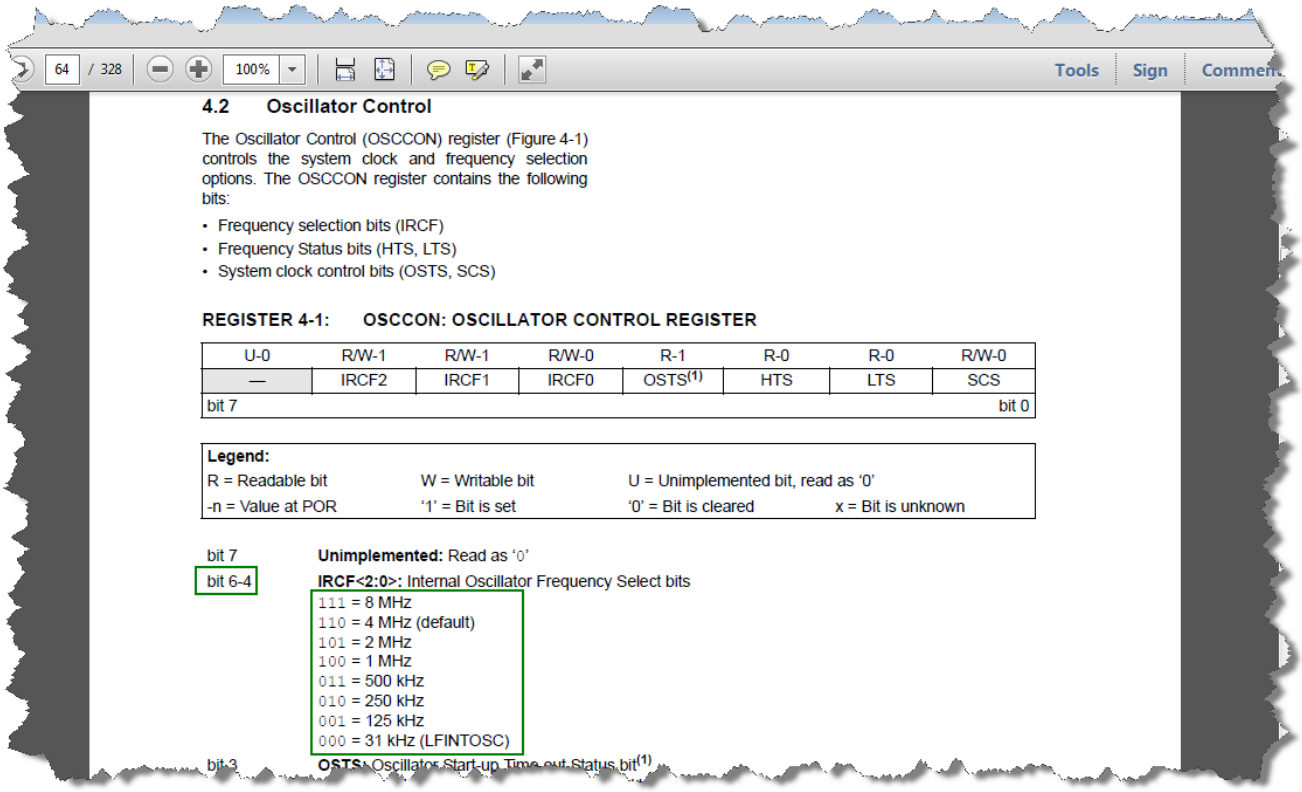
لذلك لابد من كتابة الامر التالي لتعامل مع PORTB

```
ANSELH = 0;
```

ان هذه الانواع من المايكروكنترولر تتميز بميزه جميله ولطيفه جداً ..... وهي انها تحتوي على متذبذب داخلي وهذا بتأكيد ميزه رائعه .... فهذا يريحك من شراء الكرستاله وتركيبه فهي متواجده داخلياً ولكنها بترددات محدده اذ انها لا تحتوي على ترددات عاليه .... ونحن نعلم اذا اردنا ان نستخدم مايكروكنترولر لنظام يتميز بسرعة في الاستجابه لابد من استخدام كرستاله ذات تردد عالي (20MHz) ولكن هذه السرعه غير متوفره في الكتذبذبه الداخليه لهذا المايكروكنترولر

لكن السؤال الذي يطرح نفسه الان .... ما هي السرعات أو الترددات المتواجده بداخله ... وكيف اتحكم بها؟؟؟

بسيط جداً ... افتح الداتاشيت على الصفحه ٦٤ ستكون الصفحه هي كما يلي



64 / 328 100% Tools Sign Comment

4.2 Oscillator Control

The Oscillator Control (OSCCON) register (Figure 4-1) controls the system clock and frequency selection options. The OSCCON register contains the following bits:

- Frequency selection bits (IRCF)
- Frequency Status bits (HTS, LTS)
- System clock control bits (OSTS, SCS)

REGISTER 4-1: OSCCON: OSCILLATOR CONTROL REGISTER

U-0	R/W-1	R/W-1	R/W-0	R-1	R-0	R-0	R/W-0
—	IRCF2	IRCF1	IRCF0	OSTS(1)	HTS	LTS	SCS
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **IRCF<2:0>:** Internal Oscillator Frequency Select bits
111 = 8 MHz
110 = 4 MHz (default)
101 = 2 MHz
100 = 1 MHz
011 = 500 kHz
010 = 250 kHz
001 = 125 kHz
000 = 31 kHz (LFINTOSC)

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit(1)

نلاحظ وجود الـ Oscillator Control .... يمكن التحكم بالسرعه من خلال ضبط هذا المسجل ... وخاصه الـ Bit من ٦-٤ .... اذا نختار منهم ما نشاء للحصول على السرعه المراده .... وفي المربع كما ترى ان هناك قائمه بالسرعات المتاحه .... وطريقة ضبط الـ Bit للحصول على السرعه المطلوبه

فمثلاً لو اردنا الحصول على السرعه 1MHz نكتي الجمل التاليه:

```
OSCCON.f4=1;

OSCCON.f5=0;

OSCCON.f6=0;
```

اعتقد ان الامر واضح الان .....

تمرين : افتح الداتاشيت للمايكروكنترولر PIC16f628 فهو ايضاً يحتوي على متذبذب داخليه .... واخرج قيم السرعات المتاحه من هذا المايكروكنترولر .... وقارن بينها وبين المايكروكنترولر الموجود في الاعلى (الصفحه السابقه).... فأيهما افضل من ناحية المتذبذب الداخلي .

خلاصـة

ما أريد ان تفهمه عزيزي القارئ ان مرجعك الرئيسي لفهم مبدأ عمل أي قطعه الكترونيه هو الداتاشيت وبشكل عام في بداية الداتاشيت يكون هناك وصف عام للقطعه الالكترونيه ومميزاتها وبعد ذلك تبدأ بعرض البنيه الداخليه أو الـ Block diagram للقطعه الالكترونيه وبعد ذلك بتأكيد تنطلق الى الشكل الفيزيائي لهذه القطعه وتوزع الاقطاب عليها وأخيراً تبدأ المواصفات الكهربائيه وتحليلات البنيه الداخليه .... الخ

والان وبعد ان تعلمنا بعض الامور الاساسيه في التعامل مع انواع مختلفه من المايكروكنترولر... سننتقل الى الجزء الثاني من المحاضره والذي سيكون عن بعض الطرق والافكار الاحترافية

ان أكثر ما يواجهه المبرمج في كثير من الحالات وخاصة اثناء اعداده لمشروع ضخم يحتاج الى العديد من الاطراف لتعدد الاجهزه التي سيتم التحكم بها وأيضاً تواجد عدد كبير من الحساسات ... أو الحاجه لوجود عدد كبير من ازرار التحكم التي من خلالها سيتم التحكم بالنظام ككل ..... وهذا بحد ذاته تحدي كبير بالنسبه للمهندس الكهربائي

ومن المعروف ان افضل تصميم لنظام التحكم هو الاكثر استقراراً ... بحيث لا يتأثر بتشويش الخارجي .... كما انه من الافضل ان يتم تصميمه وانشاءه بأقل مبلغ من المال

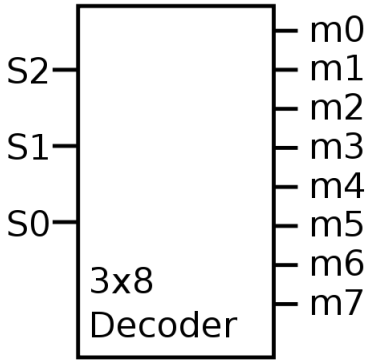
لذلك الكثير من المهندسين لا يلجئون الى المتحكمات البسيطة لعمل أنظمة التحكم بالرغم من ان النظام الذي يعملون به هو بسيط جداً ... لكنهم وللأسف الشديد يستخدمون تقنيات عاليه ومتحكمات غاليه الثمن ويدعون ان هذا هو الحل لأستقرار نظام التحكم ..... وكل هذا لسبب بسيط وهو عدم معرفتهم بأمور وتفاصيل ربما تكون بسيطه ولكن لها تأثير كبير على استقرار نظام التحكم الذي قمت بأنشاءه .

لكن الان سنتطرق الى الاجابة على سؤال واحد يحمل العديد من الاجابات .... والسؤال هو :

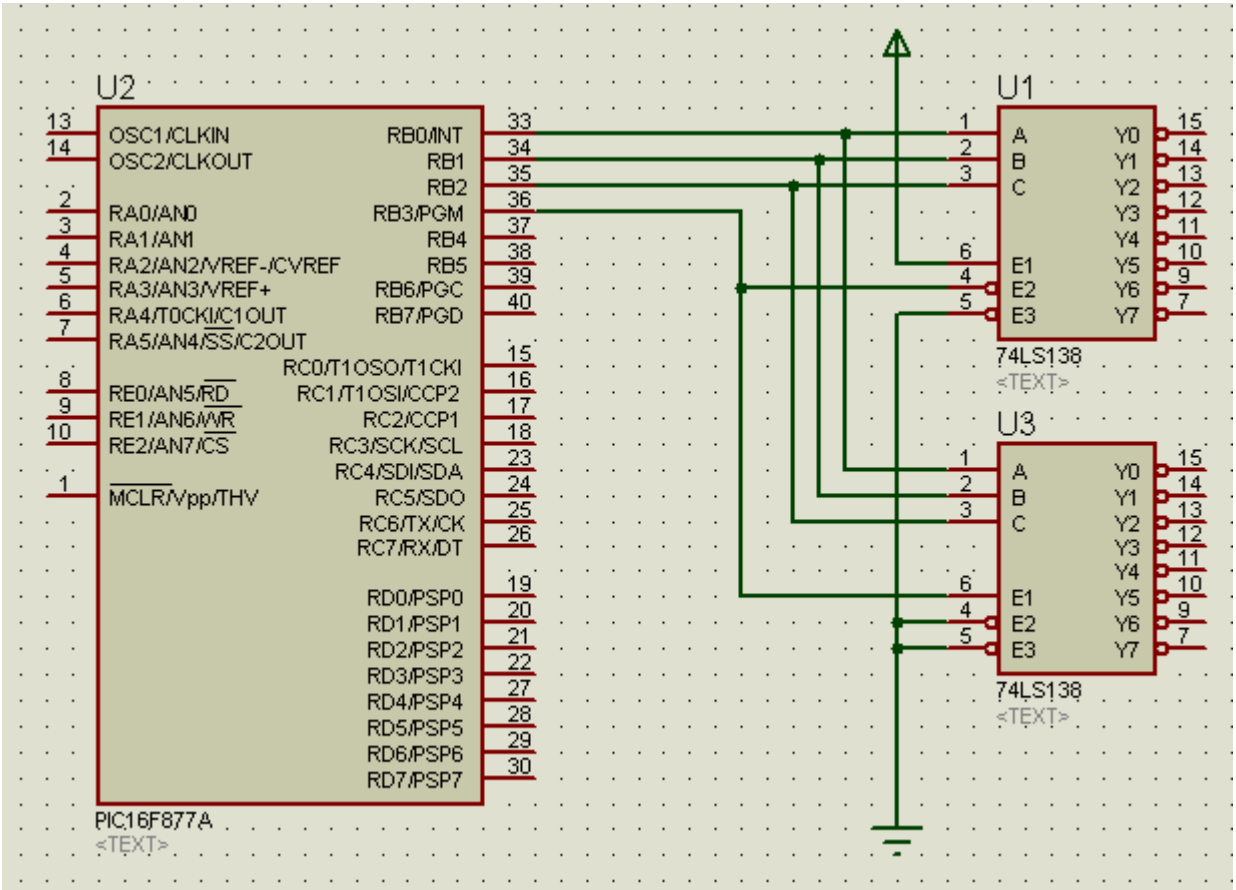
كيف يمكننا زياده اطراف التحكم في المايكروكنترولر؟؟؟

يوجد العديد من الاجابات لهذا السؤال وهنا سنتحدث عن بعضها

بتأكيد الجميع منا له معرفة بـ Decoder الموضح بالصورة التاليه .... هل يمكن استخدامه لزيادة مخارج المايكركنترولر



اذا فيمكن استغلال ثلاث PIN فقط لتحكم في 8 أجهزة وبتأكيد هذا شيء رائع انظر الى الدائرة التالية والتي تبين طريقة التوصيل البسيطه .... ولكن لا شك بأن لها أهميه لطيفة في زيادة مخارج المايكروكنترولر



لاحظ اننا استغلينا PIN 4 فقط لتحكم في 16 مخرج (16 جهاز كهربائي) .... حاول ان تقوم بعمل برنامج ليتحكم في ليدات بحيث كل ليد يعبر عن جهاز كهربائي

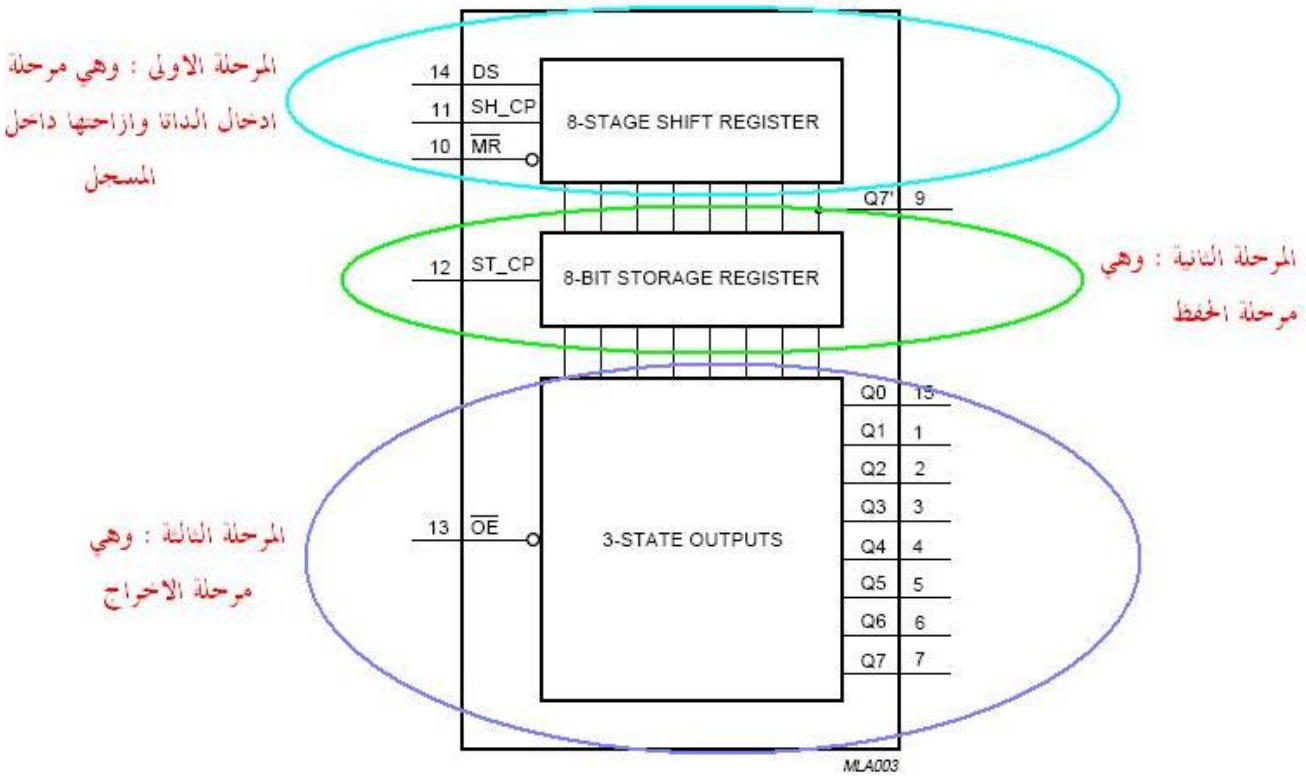
ملاحظه:

لو نفذت الدائرة السابقة ستجد شيء غريب .... فستلاحظ اننا الامر معكوس نوعا ما ... فعندما نظهر رقم ٤ مثلا على Portb ستلاحظ ان كل المخارج ستعمل ما عدا الرقم ٤ وهذا يعود الى ان الـ Decoder يعكس الحاله من 1 منطقي الى 0 منطقي فلا بد من استخدام Inverter للحصول على النتيجه الصحيحه.

وهنا ايضاً لا بد من التوجه الى طريقه ذكية هي بأستخدام المتكامله 74HC595 والتي تقوم بتحويل الاشارة التسلسلية الناتجة عن البيك الى تفرعية وبالتالي يمكن زيادة مخارج البيك بهذه الطريقة في حال كان لدينا عدد محدود من الاطراف حيث تحتاج هذه الطريقة الى ٣ اطراف فقط للحصول على عدد كبير من المخارج

تعد دارة 74HC595 من الدارات المتكاملة وهذه الدارة وظيفتها استقبال اشارة من مدخل معين واخراجها على عدة مخارج (محول من ادخال تسلسلي الى اخراج تفرعي) ..اي انها فقد تاخذ الاشارة الاولى من الطرف وتحفظها ثم تاخذ الثانية وتحفظها حتى تمتلي خانات الدارة ثم تخرجها مرة واحدة..

تتكون الدارة من ٣ مراحل موضحة في الصورة :



المرحلة الاولى: في المتكاملة هي مرحلة استقبال الداتا من الخارج عن طريق الطرف DS وتجهيزها على خطوط الداتا على خرج مخارج المرحلة الاولى (كما نلاحظ ان نهاية هذه المرحلة تتكون من ٨ خطوط داتا اي يمكن ازاحةحتى ٨ اشارات (صفر وواحد) داخل هذه المرحلة...تدعى هذه المرحلة بمرحلة الازاحة.

المرحلة الثانية: هي مرحلة الحفظ في مسجل التخزين وهي كما نرى تحفظ الاشارات ال٨ القادمة من المرحلة الاولى..

المرحلة الثالثة : تقوم باخراج القيمة الموجودة في مسجل التخزين الى الاطراف الخارجية او عدم اظهارها الى الاطراف تبعا لما هو مطلوب من الدارة القيام به.

ملاحظة مفيدة : يمكن وصل مسجلين ازاحة بحيث يكون التوصيل من مخرج المسجل الاول (output8) الى مدخل الداتا على مدخل المسجل الثاني (DATA) وهكذا نحصل على مسجل ازاحة لـ ١٦ خانة وهكذا يمكن الحصول على العدد الذي نريده من الخانات عن طريق التوصيل على التوالي للمسجلات

والان سنأخذ مثال يحدد كيفية استخدام الريجستر في اداء وظيفة محددة ... وسيتم التحكم فيه من خلال بيك 16F84A من اجل الاخراج وبالتالي عمل دارة فلاشر بسيط يستخدم هذا الريجستر.

الان برنامجنا سيكون برنامج فلاشر يستخدم البيك PIC16F84A لاخراج النبضات المطلوبة للتحكم بحيث ان الفترة الزمنية بين اضاءة كل ليد هي ثانية واحدة.

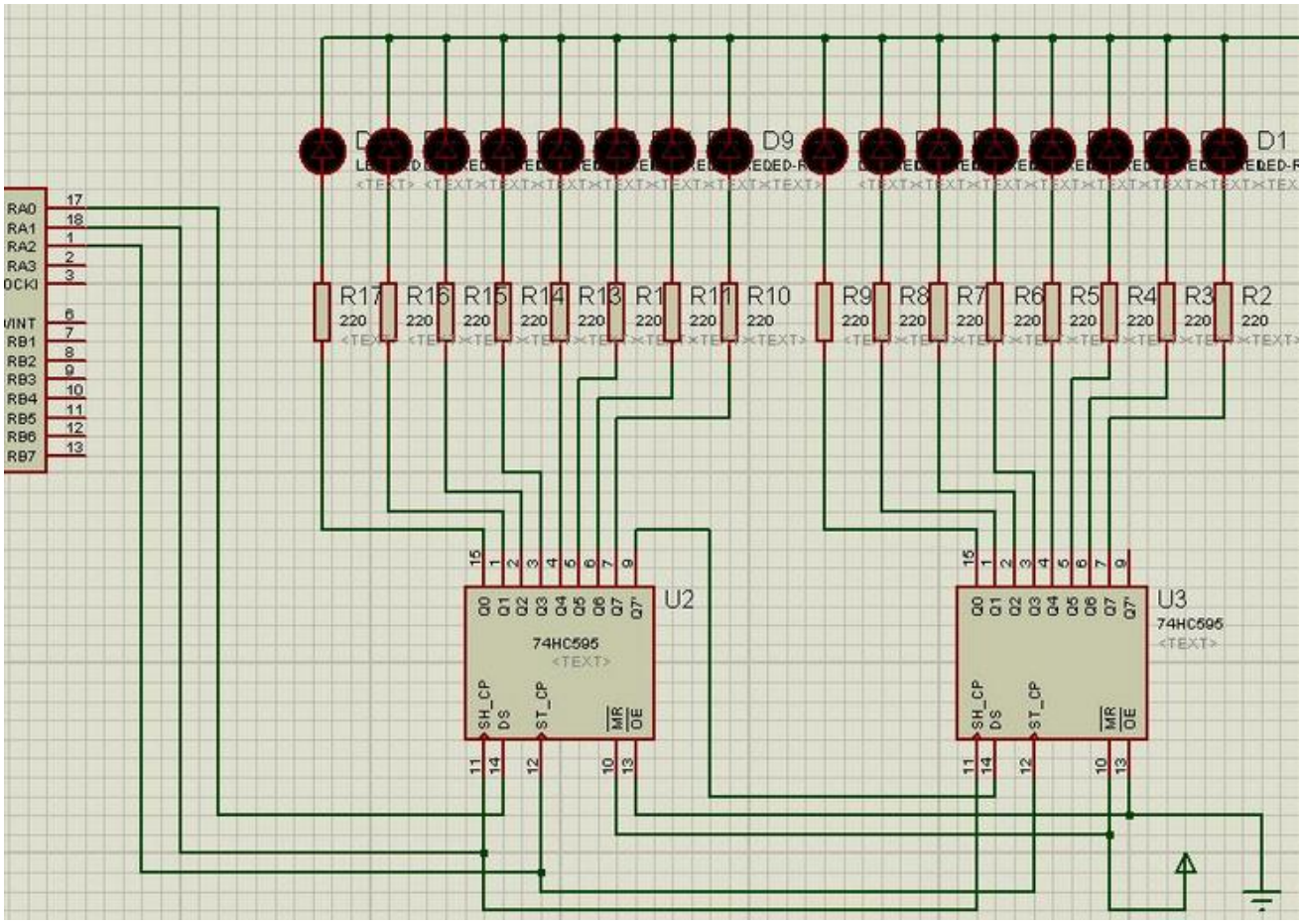
الان اريد البرنامج ان يقوم بالتالي : يقوم بتفعيل اول طرف في الريسجتر الاول وبعد ثانية واحدة ينتقل للطرف الثاني (ويطفىء الاول) وبعدها ينتقل الى الطرف الثالث ثم الرابع الى ان يصل الى الطرف رقم ١٦ (يكون الطرف الثامن في الريجستر الثاني) وبعدها يعيد نفس العملية .

الان نبدا بتحديد المطلوب من كل مرحلة من المراحل الثلاث السابقة :

١ - يجب ان يقوم بوضع الداتا (الجهد واحد منطقي على طرف الداتا DS) ونعطي نبضة ساعة ليتم ادخالها الى مسجل الازاحة (SH-CP) .وبما ان المطلوب اخراجها على الاطراف فيجب ان ننقل الداتا اللى مرحلة التخزين...
٢ -نقوم بحفظ القيمة في مسجل التخزين تحضيرا لاخراجها على الاطراف .. وذلك باعطاء نبضة ساعة للتخزين (ST-CP)
٣ - نقوم بتفعيل اخراج القيمة الموجودة في مسجل التخزين الى الاطراف وذلك بوضع القيمة (صفر منطقي) على الطرف OE.

وهكذا نكون قد عملنا ازاحة وادخلنا القيمة واحد منطقى الى الريجستر واخرجناها على الاطراف.... الان يجب ان ندخل القيمة صفر بعد الواحد ... وهكذا نقوم بعمل نفس الخطوات السابقة (١-٣) مع تغيير ان نقوم بوضع قيمة الداتا بصفر وليس واحد.... حتى نعمل ازاحة لـ١٦ مرة يجب ان تتنفذ الخطوات السابقة ١٦ مرة ...بحيث تكون اول مرة الداتا بواحد وبعدها تصبح بصفر(اول مرة تكون الداتا واحد وبعدها ١٥ مرة تكون الداتا بصفر)..

ملاحظة : بما ان ادخال الداتا Ds الى المسجل واعطاء نبضة ساعة للازاحة (SH-CP) ثم بعدها اعطاء نبضة ساعة لمسجل التخزين ..وبعدها وضع القيمة صفر على الطرف OE لاظهار القيمة على الخرج ياخذ زمن سريع نسبيا (يعني فرضا ١٠٠ ميكروثانية) وهذا لا تلاحظه العين فيمكن ان نستغني عن الطرف الواصل من البيك الى (OE) ونوصل OE مباشرة الى الارضي..وهكذا نستغل طرف من اطراف البيك لوظائف اخرى..وهكذا يصبح لدينا الدارة الموصلة في الشكل التالي:



كما نرى في الصورة فقد فان توصيل الاطراف كالتالي:

RA0-----DS

RA1-----SH-CP

RA2-----ST-CP

الان نبدا بكتابة البرنامج :

اول خطوة نضع الداتا تساوي واحد منطقي من خلال RA0 كالتالي:

```
PORTA.F0=1;
```

الان نعطي نبضة ساعة للازاحة (جبهة صاعدة) من خلال RA1 كالتالي:

```
PORTA.F1=0;

PORTA.F1=1;
```

الان نعطي نبضة ساعة للتخزين (جبهة صاعدة) وسيظهر الخرج مباشرة مع النبضة لاننا وصلنا الطرف OE الى الجهد المنخفض مسبقا وذلك من خلال الطرف RA2 كالتالي:

```
PORTA.F2=0;

PORTA.F2=1;
```

والان نعطي التاخير الزمني اللازم وهو مطلوب في السؤال (واحد ثانية) اي :

```
DELAY_MS(1000);
```

هكذا نكون اخرجنا القيمة واحد منطقي على المخرج الاول لمدة ثانية واحدة .الان نقوم بنقل الواحد الى المخرج الثاني وذلك بعملية ازاحة وادخال صفر مكانه .. وهي نفس الخطوات السابقة مع اختلاف بسيط

اول خطوة نضع الداتا تساوي صفر منطقي(وليس واحد منطقي) من خلال RA0 كالتالي:

```
PORTA.F0=0;
```

الان نعطي نبضة ساعة للازاحة (جبهة صاعدة) من خلال RA1 كالتالي:

```
PORTA.F1=0;

PORTA.F1=1;
```

الان نعطي نبضة ساعة للتخزين (جبهة صاعدة) وسيظهر الخرج مباشرة مع النبضة لاننا وصلنا الطرف OE الى الجهد المنخفض مسبقا وذلك من خلال الطرف RA2 كالتالي:

```
PORTA.F2=0;

PORTA.F2=1;
```

والان نعطي التاخير الزمني اللاذز حتى تلاحظة العين وهو مطلوب في السؤال (واحد ثانية) اي :

```
DELAY_MS(1000);
```

وبالتالي البرنامج يكون اول مرة لاخراج الواحد منطقي :

```
PORTA.F0=1;
PORTA.F1=0;
PORTA.F1=1;
PORTA.F2=0;
PORTA.F2=1;
DELAY_MS(1000);
```

ويمكن استخدام الحلقات للقيام بعملية الازاحه ١٥ مره مثلا حلقة FOR كالتالي:

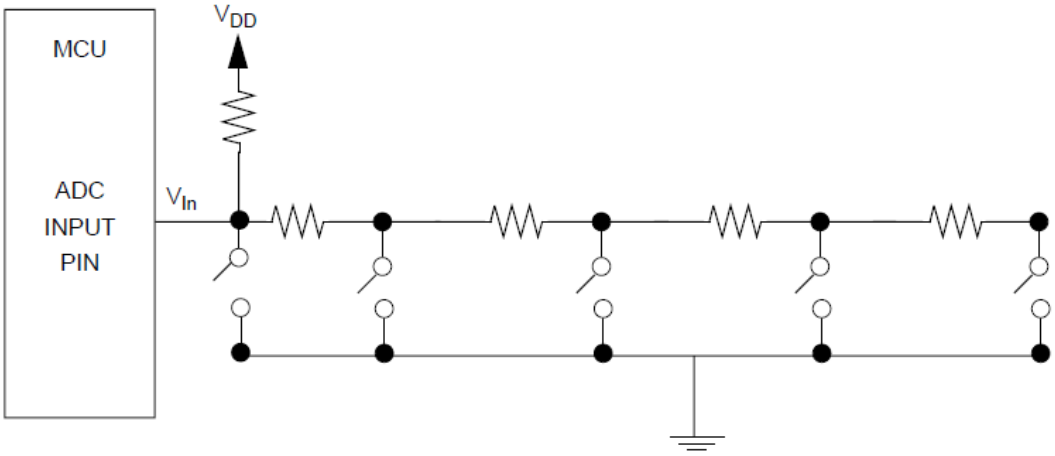
```
FOR(I=0;I<15;I++)
{
PORTA.F0=0;
PORTA.F1=0;
PORTA.F1=1;
PORTA.F2=0;
PORTA.F2=1;
DELAY_MS(1000);}
```

ولا ننس ان نضع البرنامج كله ضمن حلقة لاعادة تنفيذ البرنامج دائما .. وهكذا يكون البرنامج كاملا مع اعداد المداخل والمخارج والتغيرات له الشكل التالي:

```
Void main()
{
Int I;
TRISA=0;
Loop:
PORTA.F0=1;
PORTA.F1=0;
PORTA.F1=1;
PORTA.F2=0;
PORTA.F2=1;
DELAY_MS(1000);
FOR(I=0;I<15;I++)
{
PORTA.F0=0;
PORTA.F1=0;
PORTA.F1=1;
PORTA.F2=0;
PORTA.F2=1;
DELAY_MS(1000);
} Goto Loop;
}
```

والان لننتقل الى طريقة جميله أيضاً لزياده مداخل المايكروكنترولر .... فمثلاً هل من المكن توصيل خمسة مفاتيح Switch على مدخل واحد من مداخل المايكروكنترولر .... الجواب نعم

الن الكثير من المايكروكنترولر الان تدعم خاصية الـ ADCوالتي تحدثنا عنها في لمحاضرات السابقه ، وفي المايكروكنترولر 16f877a تخزن القيمه بريجستر 10 Bit .... والان انظر الى الدائره التاليه



لاحظ ان بتوصيل مقاومات مع المفاتيح كما في الدائره العلويه .... فأن قيمة الفولتيه التي ستدخل الى ADC تعبر عن المفتاح المضغوط وبالتالي تنفيذ الامر المراد تفيذه بهذا المفتاح

وقيمة الفولتية الداخله تكون بين high و Low .... ولو اردنا حساب دقة هذه الطريقه ببساطه نأخذ اقصى قيمة للـ high وهي 5V مقسمه على 1023 فتكون النتيجه 4.88mV وهذا يعني تغير واحد في LSB يؤدي الى تغير بمقدار 4.88mV بالفولتيه الداخله على الـ ADC .

قال رسول الله صلى الله عليه وسلم : من سئل عن علم فكتمه ألجمه الله بلجام من نار يوم القيامة